In the Claims:

Please amend the claims as indicated below.

1. (currently amended) A method for manipulating data in a processor, the method comprising:

performing a conditional shift operation on an index register based on the condition of a carry flag, the condition of the carry flag having been set by a previous arithmetic operation; ~~and~~

performing an indexed load operation using the ~~an~~ index register;

wherein the conditional shift operation and the indexed load operation are performed during execution of a load-shift carry instruction; and

implementing a binary search of keys ordered in a table by executing the load-shift carry instruction.

2. (currently amended) The method for manipulating data in a processor according to claim 1, further comprising:

transferring data from an input buffer to a packet task manager; dispatching the data from the packet task manager to an analysis machine;

classifying the data in the analysis machine; and

implementing the ~~a~~ binary search in the analysis machine.

3. (original) The method for manipulating data in a processor according to claim 1, further comprising modifying and forwarding the data in a packet manipulator.

4. (original) The method for manipulating data in a processor according to claim 2, further comprising transferring the data after modifying and forwarding to an output buffer.

5. (original) The method for manipulating data in a processor according to claim 1, further comprising processing data at a rate of at least 10 Gbs.

6. (currently amended) A processor having an instruction set associated therewith, the instruction set including a load-shift carry instruction that, when executed by the processor causes the processor to:

perform a conditional shift operation on an index register based on the condition of a carry flag, the condition of the carry flag having been set by a previous arithmetic operation; ~~and~~

perform an indexed load operation using the ~~an~~ index register; and

wherein the processor includes an analysis machine that implements a binary search of keys ordered in a table by executing the load-shift carry instruction.

7. (currently amended) A processor according to claim 6, wherein the processor comprises:

said ~~an~~ analysis machine having multiple pipelines, wherein one pipeline is dedicated to directly manipulating individual data bits of a bit field;

a packet task manager operationally connected to said analysis machine; and;

a packet manipulator operationally connected to said analysis machine.

8. (original) The processor according to claim 7, wherein said analysis machine is multi-threaded.

9. (original) The processor according to claim 7, wherein said analysis machine has 32 threads.

10. (original) The processor according to claim 7, further comprising: a packet task manager operationally connected to said analysis machine; a packet manipulator operationally connected to said analysis machine; and a global access bus including a master request bus and a slave request bus separated from each other and pipelined.

11. (original) The processor according to claim 7, further comprising: an external memory engine operationally connected to said analysis machine; and a hash engine operationally connected to said analysis machine.

12. (original) The processor according to claim 10, further comprising: packet input global access bus software code used for flow of data packet information from a flexible input data buffer to an analysis machine.

13. (original) The processor according to claim 10, further comprising: packet data global access bus software code used for flow of packet data between a flexible data input bus and a packet manipulator.

14. (original) The processor according to claim 10, further comprising: statistics data global access bus software code used for connection of an analysis machine to a packet manipulator.

15. (original) The processor according to claim 10, further comprising: private data global access bus software code used for connection of an analysis machine to an internal memory engine submodule.

16. (original) The processor according to claim 10, further comprising: lookup global access bus software code used for connection of an analysis machine to an internal memory engine submodule.

17. (original) The processor according to claim 10, further comprising: results global access bus software code used for providing flexible access to an external memory.

18. (original) The processor according to claim 10, further comprising: results global access bus software code used for providing flexible access to an external memory.

19. (original) The processor according to claim 10, further comprising: a bidirectional access port operationally connected to said analysis machine; a flexible data input buffer operationally connected to said analysis machine; and a flexible data output buffer operationally connected to said analysis machine.